

(11)特許出願公開番号

特開平8-185180

(43)公開日 平成8年(1996)7月16日

(51) Int. Cl. ⁸	識別記号	序内整理番号	F I	技術表示箇所
G 1 0 H 1/38	Z			
1/00	1 0 2 B			

審査請求 未請求 請求項の数1 OL (全 13 頁)

(21)出願番号 特願平6-328824

(22)出願日 平成6年(1994)12月28日

(71)出願人 000004075

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(72)発明者 近藤 昌夫

静岡県浜松市中沢町10番1号 ヤマハ株式会社
会社内

(72)發明者 中國 裕樹

静岡県浜松市中沢町10番1号 ヤマハ株式会社
会社内

(72)発明者 伊藤 真一

静岡県浜松市中沢町10番1号 ヤマハ株式会社
社内

(74)代理人 弁護士 瀧野 秀雄 (外1名)

(54) 【発明の名称】 自動伴奏装置

(57) 【要約】

【目的】自動伴奏装置において、伴奏パターンの種類に応じた各ソースパターン毎にノート変換テーブルを作成する手間を省くとともに、音域制限を無くす。

【構成】和音の種類に対応するスケール（音階）に基づいて、和音の種類に対する音名の属性として、コード音（c1～c5）、スケール音（s）、ノンスケール音（n）を分類テーブルに記憶しておく。各属性に対応するノート変換テーブルを記憶しておく。ソースパターンにおけるキーコードを、その音名とソースパターンの和音の種類に基づいて属性で分類する。分類された属性に対応するノート変換テーブルでそのキーコードを音高変換する。指定和音の根音とソースパターンの和音の根音との差が大きいときは、分類した属性を変換し、変換された属性に対応するノート変換テーブルで、指定和音における属性が基伴奏パターンの和音における属性に近かつ異なるキーコードとなるように音高変換する。

ATBL (STP, NT) 分類テーブル

STP ↓		音名 (NT)											
		C	C#	D	D#	E	F	F#	G	G#	A	A#	B
和音種類	1	C#m	c4	n	s	n	c2	n	n	c5	n	n	n
										
	...	C#m7(9)	c4	n	c1	n	c2	n	n	c5	n	n	n
										

1

【特許請求の範囲】

【請求項1】 和音に応じたスケールに対する音名の属性に基づいて、基伴奏パターンのキーコードを該基伴奏パターンの和音と入力される指定和音とに応じたキーコードに音高変換して自動伴奏を行う自動伴奏装置であって、
前記基伴奏パターンを供給する基伴奏パターン供給手段と、
前記基伴奏パターンの和音を供給する和音供給手段と、
前記指定和音を入力する和音入力手段と、
前記基伴奏パターン供給手段から供給される基伴奏パターンのキーコードについて前記和音供給手段から供給される和音に応じた該キーコードの前記属性を検出する属性検出手段と、
前記和音入力手段で入力された指定和音の根音と前記基伴奏パターンの和音の根音との差が予め定められた範囲以上のときに、前記基伴奏パターンのキーコードを、指定和音における属性が基伴奏パターンの和音における属性に近かつ異なるキーコードとなるように音高変換する変換手段と、
前記音高変換されたキーコードに応じた伴奏音を生成する伴奏音生成手段と、を備えたことを特徴とする自動伴奏装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、電子楽器等に備えられ、予め記憶された伴奏パターンに基づいて演奏者の演奏に合わせて自動伴奏を行う自動伴奏装置に関する。

【0002】

【従来の技術】従来、自動伴奏装置は、例えばロックやカントリーなど楽曲の各種スタイル（ジャンル等）に応じた多種類の伴奏パターンを記憶しており、自動伴奏時には、演奏者の選択した伴奏パターンに基づいて、鍵盤等から検出した指定和音に合わせて自動伴奏を行う。

【0003】ここで、伴奏パターンは、個々の伴奏音の音高に応じた音高データを発音タイミングを示すタイミングデータとセットにしたものであるが、この音高データとしては例えばC Majなど所定和音に基づくキーコードが用いられ、指定和音の種類と根音に応じた音高変換を考慮したソースパターンとして作成される。

【0004】すなわち、自動伴奏時には、鍵盤等から入力される指定和音の種類に応じてソースパターンのキーコードを音高変換するとともに、全音高を指定和音の根音に応じてシフトし、和音に合った音高の伴奏音を生成している。

【0005】なお、所定和音で作成したソースパターンを指定和音の種類に合ったように音高変換するために、ソースパターンのキーコードについてのシフトデータを和音の種類に対応付けてノート変換テーブルとして記憶しておき、入力される和音の種類に応じたシフトデータ

2

をノート変換テーブルから読み出し、この読み出したシフトデータをソースパターンのキーコードと演算することにより和音の種類に応じた伴奏パターンを得るようにしている。

【0006】

【発明が解決しようとする課題】しかし、多くの伴奏パターンは装飾的な音など和音構成音以外の音を含んでおり、最適な伴奏パターンを得るためには、これらの和音構成音以外の音に対して伴奏パターンの種類に応じた音高変換を行う必要がある。このため、パターン作成者は、音高変換を行うためのノート変換テーブルをソースパターン毎（伴奏のスタイル毎）に作成する必要があった。

【0007】また、ソースパターンのキーコードを単に指定和音の根音に基づいて音高変換するだけでは、変換後のキーコードが本来の音域から外れてしまうことがあるので、音域制限をする必要がある。なお、この音域制限では例えば高すぎる音をオクターブ下げるなど行っているが、このようにすると、テンションノートがベース音の領域になるなどして和音の響きが変わってしまうという問題がある。

【0008】本発明は、伴奏のスタイルに応じた各ソースパターン毎にノート変換テーブルを作成する必要がなく、音域制限を必要としない自動伴奏装置を得ることを課題とする。

【0009】

【課題を解決するための手段】上記の課題を解決するためになした本発明の自動伴奏装置は、和音に応じたスケールに対する音名の属性に基づいて、基伴奏パターンのキーコードを該基伴奏パターンの和音と入力される指定和音とに応じたキーコードに音高変換して自動伴奏を行う自動伴奏装置であって、前記基伴奏パターンを供給する基伴奏パターン供給手段と、前記基伴奏パターンの和音を供給する和音供給手段と、前記指定和音を入力する和音入力手段と、前記基伴奏パターン供給手段から供給される基伴奏パターンのキーコードについて前記和音供給手段から供給される和音に応じた該キーコードの前記属性を検出する属性検出手段と、前記和音入力手段で入力された指定和音の根音と前記基伴奏パターンの和音の根音との差が予め定められた範囲以上のときに、前記基伴奏パターンのキーコードを、指定和音における属性が基伴奏パターンの和音における属性に近かつ異なるキーコードとなるように音高変換する変換手段と、前記音高変換されたキーコードに応じた伴奏音を生成する伴奏音生成手段と、を備えたことを特徴とする。

【0010】

【作用】本発明の自動伴奏装置は、和音に応じたスケールに対する音名の属性に基づいて、基伴奏パターンのキーコードを該基伴奏パターンの和音と入力される指定和音とに応じたキーコードに音高変換して自動伴奏を行

う。このとき、指定和音の根音と基伴奏パターンの和音の根音との差が予め定められた範囲以上のときは、基伴奏パターンのキーコードを、指定和音における属性が基伴奏パターンの和音における属性に近かつ異なるキーコードとなるように音高変換する。

【0011】前記和音に応じたスケールに対する音名の属性は、基伴奏パターンの種類に係わらず音楽的に最適なものとして決められており、基伴奏パターンのキーコードを基伴奏パターンの和音の属性に基づいて指定和音に応じて音高変換すると、そのキーコードは基伴奏パターンの種類に係わらず、音楽的に最適なものに変換できる。また、指定和音の根音と基伴奏パターンの根音とが大きく離れている場合は、基伴奏パターンのキーコードは、指定和音における属性が検出された属性に近かつ異なるキーコードに変換されるので、音域制限をしなくても上記根音の差が変換後のキーコードの音域をシフトさせることが無くなる。

【0012】

【実施例】図1は本発明実施例の自動伴奏装置を適用した電子楽器のブロック図であり、CPU1はプログラムメモリ2に格納されている制御プログラムに基づいてワーキングメモリ3のワーキングエリアを使用して電子楽器全体の制御を行い、鍵盤4の操作による鍵盤演奏、パネルスイッチ等の操作子5の操作によるモードの切り換えやデータの入力処理、リズムパターンメモリ6および伴奏パターンメモリ7に記憶されている伴奏パターンに基づく自動伴奏を行う。

【0013】具体的には、CPU1は鍵盤4のキーイベントを検出し、キーイベントのあった鍵に対応するキーコードをキーオン信号またはキーオフ信号とともに取り込み、音源8にキーコードとノートオンまたはノートオフを出力して鍵盤演奏に対応する発音処理と消音処理を行う。なお、音源8は入力されるキーコードおよびリズムパターンの伴奏音を発音するための打楽器番号に応じた楽音を発生し、この楽音信号はサウンドシステム9に出力され、サウンドシステム9はD/A変換、増幅等を行って楽音を発生する。

【0014】また、CPU1は操作子5の操作によって指定されたテンポの情報をタイマ10に設定し、タイマ10はCPU1から設定されたテンポに応じて8分音符1個毎に12回(96分長に1回)の割込み信号を発生し、CPU1はこの割込み信号により割込み処理を行って自動伴奏の処理を行う。

【0015】さらに、鍵盤4は低音側の左鍵域と高音側の右鍵域とに仮想的に分割されており、自動伴奏時には、CPU1は、右鍵域のキーイベントに対して発音処理と消音処理を行い、左鍵域のキーイベントに対しては検出されたキーコードに基づいて和音を検出する。

【0016】操作子5は、自動伴奏を行う楽曲のスタイルを選択するためのスタイル選択スイッチ、自動伴奏のスタートとストップを指示するためのスタート/ストップスイッチ、ユーザーによる伴奏パターンおよびリズムパターンの入力モードを指定するパターン入力スイッチ、およびその他のスイッチを備えており、CPU1は、操作子5の各スイッチの操作イベントを検出して操作イベントのあったスイッチに対応する処理を行う。

【0017】また、リズムパターンメモリ6と伴奏パターンメモリ7には、図示しないメモリにプリセットされている多種類のリズムパターンと基伴奏パターンとしての伴奏パターン(ソースパターン)が記憶され、さらに、伴奏パターン入力モードにより入力されリズムパターンと伴奏パターンが記憶される。なお、これらのリズムパターンと伴奏パターンはスタイル選択スイッチで選択されてパターン番号(PTN)によって記憶される。

【0018】この実施例における伴奏パターンは、アルペジオ等の各種のコード音に対応するコードバックング1~3の3つのパートからなるコードパターンと、ベース音に対応する1つのベースパターンとで構成され、これらのコードバックング1~3およびベースパターンは、それぞれパート番号(PRT=0~3)によって指定される。

【0019】なお、各伴奏パターンは、各パート毎にキーコード、タイミング、キーオンまたはキーオフのデータをセットにして順次記憶したものであり、リズムパターンは打楽器番号とタイミングのデータをセットにして順次記憶したものである。

【0020】ところで、次表1に示したように、和音の種類に対して取り得るスケール(音階)はそれぞれ決まっており、これらのスケールの各々について、和音構成音はコード音(c)、和音構成音以外でスケール上の音はスケール音(s)、スケールから外れた音はノンスケール音(n)として、音名(C, C#, D, D#, ...)に対して属性(c, s, n)が音楽的に決まっている。

【0021】

【表1】

和音種類名	スケール名	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Ma j	Ionian	c	n	s	n	c	s	n	c	n	s	n	s
	Lydian	c	n	s	n	c	n	s	c	n	s	n	s
	Mixolydian	c	n	s	n	c	s	n	c	n	s	s	n
	result	c	n	s	n	c	n	n	c	n	s	n	n
⋮	⋮	⋮											
7-9	Mixolydian	c	n	c	n	c	s	s	c	s	s	c	n
	Aeolian	c	n	c	c	n	s	n	c	s	n	c	n
	result	c	n	c	n	n	s	n	c	s	n	c	n

【0022】そこで、自動伴奏のソースパターンのキーコードを、ソースパターンの和音に対応させてこれらの属性で分類し、各属性に応じたノート変換テーブルにより指定和音においても同じ属性になるように音高変換を行うと、伴奏パターンの種類に係わりなく音楽的に最適な音高変換を行うことができる。

【0023】ただし、表1からわかるように、同じ和音の種類で同一音名でもスケールの違いによって属性が異なるものがあるので、各音名に対して、同じ和音の種類の中の全てのスケールで同一の属性となる場合はその属性とし、同じ和音の種類の中の全てのスケールで複数の属性となる場合は、1つでもノンスケール音(n)となるものがある場合はノンスケール音とする。なお、コード音とスケール音の混在はありえない。

【0024】このような条件で属性を決定すると、各和音の種類に対する各音名の属性は表1の「result」の欄のようになり、和音の種類と音名に対して1つの属性を対応させた分類テーブルが得られる。

【0025】図2は上記の条件に従って作成した実施例における分類テーブルの一例を示す図であり、この分類テーブルは、各和音の種類(TP)に対して各音名(C, C#, D, D#, ...)がコード音(c1~c5)、スケール音(s)およびノンスケール音(n)の何れの属性を有するかを分類したものである。

【0026】また、この実施例では、コード音の属性については、コードバックキングを生成するための重要度を考慮して“c1~c5”のように重要度の高い順に同類の属性として分類するようにしている。なお、コード音の記号(c)に付した数字は、コード音の属性の中でもコードバックキングを生成するために重要な音名ほど小さな値をとる。このうち、“c1, c2, c3”は和音の特徴を表す重要な属性であり、4音コードや5音コードの場合のテンションノートはこの“c1, c2, c3”のどれかに割り当てられている。また、1度および5度のノートは“c4, c5”に割り当てられている。

【0027】なお、このような、和音、属性および音名の関係は、和音の種類TPとノートコードNT(音名に*

*対応するコード)を引数とする配列レジスタに各属性(c1~c5, s, n)を示すデータを格納することにより、分類テーブルATBL(TP, NT)として分類テーブルメモリ11に記憶されている。

【0028】そして、自動伴奏時には、ソースパターンから読み出したキーコードをノートコードNTに変換し、このノートコードNTとソースパターンの和音の種類STPから属性ATBL(STP, NT)を求め、この属性に対応するノート変換テーブルを選択する。そして、このノート変換テーブルにより、属性が“c1~c5”のものは指定和音における属性が“c1~c5”となるように、また、属性が“s”のものは指定和音における属性が“s”となるように、さらに、属性が“n”のものは指定和音における属性が“n”となるようにキーコードを変換し、このキーコードに基づいて伴奏音を発生する。

【0029】ただし、属性が“c1~c5”のものについては、変換後のキーコードがコード音としてふさわしい音域となるように、指定和音の根音がソースパターンの和音の根音よりも所定範囲以上離れているときは、この同類の属性“c1~c5”の中で属性を変換する。このために、プログラムメモリ2には、ソースパターンの和音の根音と指定和音の根音との差分に応じて上記コード音の属性を変換する属性変換テーブルが記憶されている。

【0030】図3は実施例の属性変換テーブルを示す図である。なお、この属性変換テーブルは、根音の差分Lとコード音の属性(cj)の添字jを引数とする配列レジスタに各属性(c1~c5)に対応する添字jのデータを格納することにより、属性変換テーブルSFL(j, L)としてプログラムメモリ2に記憶されている。

【0031】この属性変換テーブルは、図3(A)のように属性“1, 2, 3”(c1, c2, c3)の組を一つの群として変換し、図3(B)のように属性“4, 5”(c4, c5)の組を一つの群として変換する。なお、このように2つの群に分けて扱うのは、属性“c1, c2, c3”の音は和音の特徴を表す重要な属性であるの

でこれらの音を属性“c4, c5”の1度や5度と分離することにより和音の特徴の再現性を保持するためである。また、属性“c1, c2, c3”のテンションノートが1度や5度に置き変わって低い音となり、和音の響きが変わってしまうのを防止するためである。なお、この属性変換は全てのソースパターンについて行うものではなく、各ソースパターン毎に属性変換を行うか否かを示すフラグが設定されている。

【0032】上記の各群において属性の変換を次のように変換する。すなわち、属性“1, 2, 3”については、根音の差分Lが“-1~2”の場合は恒等置換とし、Lが“3~6, -6”の場合は“1”を“3”に、“2”を“1”に、“3”を“2”にそれぞれ変換する。また、Lが“-5~-2”の場合は“1”を“2”に、“2”を“3”に、“3”を“1”にそれぞれ変換する。属性“4, 5”については、差分Lが“-2~3”の場合は恒等置換とし、Lが“4~6, -6~-3”の場合は“4”を“5”に、“5”を“4”にそれぞれ変換する。

【0033】図4は属性変換の一例を示す図である。この例は、CMaj9のソースパターンを種類がMaj9の指定和音に音高変換する場合の属性変換の例を示しており、図4(A)は属性変換前のノートコードと属性の関係を示し、図4(B)は属性変換後のノートコードと属性の関係を示している。

【0034】図5は実施例におけるノート変換テーブルの一例を示す図である。このノート変換テーブルは、属性に応じたインデックス(AT)、和音の種類(TP)、ノートコード(NT)を引数とする配列レジスタにキーコードのシフトデータ(0, -1, -2, ...)を記憶したもので、ノート変換テーブルNTT(AT, TP, NT)としてノート変換テーブルメモリ12に記憶されている。

【0035】なお、図5(A)は属性がコード音でコードバックの音高変換に用いるノート変換テーブルの一例、図5(B)は属性がコード音でベースの音高変換に用いるノート変換テーブルの一例、図5(C)は属性がスケール音のノート変換テーブルの一例、および、図5(D)は属性がノンスケール音のノート変換テーブルの一例を示している。なお、図5(A)のノート変換テーブルは音を下にシフトする場合は最大で半音4つで、音を上にシフトする場合は最大で半音7つとなっている。

【0036】以上の構成により、自動伴奏時に、CPU1は現在選択されているソースパターンのキーコードを伴奏パターンメモリ7から読み出し、このキーコードに対応するノートコードにより分類テーブルメモリ11の分類テーブルを参照してこのキーコードをソースパターンの和音の種類に応じてコード音、スケール音および非スケール音の何れかの属性に分類する。そして、ソースパターンの根音と指定和音の根音との差分が所定値以上

であれば属性を属性変換テーブルで変換し、これらの属性と指定和音とに応じてノート変換テーブルメモリ13に記憶されているノート変換テーブルを参照してキーコードを音高変換する。

【0037】図6は実施例における制御プログラムのメインルーチン、図7は割込み処理ルーチンのフローチャート、図8および図9はサブルーチンのフローチャートであり、各フローチャートに基づいて実施例の動作を説明する。なお、以下の説明およびフローチャートにおいて、制御に用いられるレジスタおよびフラグを下記のラベルで表記し、各レジスタおよびフラグとそれらの記憶内容は特に断らない限り同一のラベルで表す。

【0038】RT: 指定和音の根音のレジスタ

TP: 指定和音の種類のレジスタ

SRT(i): パターン番号iのソースパターンの和音の根音のレジスタ

STP(i): パターン番号iのソースパターンの和音の種類のレジスタ

RUN: 自動伴奏のスタート/ストップを示すフラグ

PTN: 伴奏パターンおよびリズムパターンのパターン番号のレジスタ

PRT: 伴奏パターンのパートを示すパート番号のレジスタ

KC: キーコードのレジスタ

NT: ノートコードのレジスタ

ATBL(k, m): 分類テーブル

ATRB: 分類された属性のレジスタ

j: コード音の属性のうちの種類を区別する添字のレジスタ

SFLF(PTN): パターン番号PTNのパターンについて属性変換をするか否かを示すフラグ

L: ソースパターンの和音の根音と指定和音の根音の差分のレジスタ

SFL(j, L): 属性変換テーブル

NTT(p, k, m): ノート変換テーブル

AT: 属性とノート変換テーブルを対応させるインデックスのレジスタ

D: ノート変換テーブルのシフトデータのレジスタ

【0039】まず、電源の投入等によってCPU1が図6のメインルーチンの処理を開始すると、ステップS1で各フラグおよびレジスタのリセット等の初期設定を行い、ステップS2で鍵盤4におけるキーイベントの有無を判定し、キーイベントが無ければステップS6に進み、キーイベントが有ればステップS3に進む。

【0040】ステップS3では、「イベントキーが左鍵域でかつRUN=1」であるか否かを判定し、「イベントキーが左鍵域でかつRUN=1」でなければステップS4でイベントキーについての発音または消音処理を行ってステップS6に進む。「イベントキーが左鍵域でかつRUN=1」であれば、ステップS5でイベントキー

のキーコードに基づいて和音を検出し、指定和音の根音をレジスタRTに格納するとともに指定和音の種類をレジスタTPに格納してステップS6に進む。

【0041】ステップS6では、操作子5におけるスタイル選択スイッチのオンイベントの有無を判定し、オンイベントが無ければステップS8に進み、オンイベントが有ればステップS7で、選択されたスタイルに対応するパターン番号をレジスタSTNに格納し、ステップS8に進む。

【0042】ステップS8では、操作子5におけるスタート/ストップスイッチのオンイベントの有無を判定し、オンイベントが無ければステップS12に進み、オンイベントが有れば、ステップS9でフラグRUNを反転してステップS10でRUN=1となったか否かを判定する。RUN=1となっていなければ(RUN=0)、自動伴奏状態でストップが指示されたことになるのでステップS12に進み、RUN=1となっていれば自動伴奏のスタートが指示されたことになるので、ステップS11でタイミングクロックをクリアしてステップS12に進む。そして、ステップS12では、ユーザによる伴奏パターン、リズムパターンおよび和音情報の入力処理や音色の選択などその他の処理を行い、各処理が終了するとステップS2に戻ってステップS2以降の処理を繰り返す。

【0043】図7の割込み処理は、タイマ10からの割込み信号により96分長毎に起動され、先ずステップS21でRUN=1であるか否かを判定する。RUN=1でなければ元のルーチンに復帰し、RUN=1であれば、ステップS22で、パターン番号PTNのリズムパターン中の現在のタイミングクロックに対応するデータ(楽器番号)を読み出し、音源8に出力して再生する。

【0044】次に、ステップS23で、伴奏パターンについて再生するパートのパート番号PRTを“0”にセットし、ステップS204でのPRTのインクリメントとステップS205の判定処理により、コードバックグ1~3およびベースパターンの各パートについてステップS24以降の処理を繰り返す。

【0045】ステップS24では、伴奏パターンメモリ12から、パターン番号PTNの伴奏パターンのパートPRT中で現在のタイミングクロックに対応するデータを読み出し、ステップS25で再生データがあるか否かを判定し、再生データが無ければステップS204に進み、再生データがあればステップS26に進む。

【0046】ステップS26では、再生データがキーオンデータであるか否かを判定し、キーオンデータでなければ、ステップS27でキーオフ信号をPRTに対応するチャンネル番号と共に音源8に出力して消音処理を行い、キーオンデータであれば、ステップS28で発音キーコードすなわち伴奏パターンの読出しデータであるキーコードをレジスタKCに格納し、ステップS29でバ

ート番号PRT=3(ベースパターンのパート)であるか否かを判定する。

【0047】PRT=3でなければコードパターン(コードバックグ1~3:PRT=0, 1, 2)の再生であるので、ステップS201で図8のコードパターンについての音高変換処理を行い、PRT=3であればベースパターンの再生であるので、ステップS202で図9のベースパターンについての音高変換処理を行う。

【0048】後述説明するようにコードパターンまたはベースパターンについての音高変換処理が終了すると、ステップS203で、キーオン信号および音高変換されたキーコードKCを、PRTに対応するチャンネル番号と共に音源8に出力し、発音処理を行う。

【0049】そして、ステップS204でPRTをインクリメントしてステップS205でPRT=4であるか否かを判定し、PRT=4でなければ伴奏パターンの再生パートが残っているのでステップS24以降の処理を行い、PRT=4であれば全てのパートを再生したのでステップS206でタイミングカウンタでカウントを行って元のルーチンに復帰する。

【0050】図8のコードパターンについての音高変換処理では、先ず、ステップS31で、読出しデータのキーコードKCをパターン番号PTNの伴奏パターンの和音の根音SRT(PTN)で音高シフトして、そのノートコードをレジスタNTに格納し、ステップS32で、パターン番号PTNの伴奏パターンの和音の種類STP(PTN)と発音ノートコードNTとに対応する分類テーブルのデータATBL(STP(PTN), NT)をレジスタATRBに格納する。

【0051】次に、ステップS33で、ATRB=cj(コード音)であるか否かを判定し、ATRB=cjでなければ発音ノートコードNTの属性がスケール音かノンスケール音であるのでステップS34に進み、ステップS34で、ATRB=s(スケール音)のときATに“6”をセットし、ATRB=n(ノンスケール音)のときATに“7”をセットし、ステップS39に進む。

【0052】ステップS33でATRB=cjであれば発音ノートコードNTの属性がコード音であるのでステップS35に進み、ステップS35で、SFLF(PTN)=1であるか否かを判定し、SFLF(PTN)=1でなければ属性変換を行わないソースパターンであるので、ステップS38で、レジスタATに“j-1”をセットしてステップS39に進む。

【0053】ステップS35でSFLF(PTN)=1であれば、属性変換の対象となるソースパターンであるので、ステップS36で、指定和音の根音RTとソースパターンの和音の根音SRT(PTN)の差分を求めてレジスタLに格納し、ステップS37で、求めた属性の添字jと根音の差分Lに基いた属性変換テーブルから変換後の属性の添字SFL(j, L)を読み出してレジス

タjに格納し、ステップS38で、レジスタATに“j-1”をセットしてステップS39に進む。

【0054】ステップS39では、読出しデータのキーコードKCを指定和音の根音RTで音高シフトしてそのノートコードをレジスタNTに格納し、ステップS301で、ATに応じて選択したノート変換テーブルから、指定和音の種類TPと上記ノートコードNTに対応するシフトデータNTT(AT, TP, NT)を読み出し、これをレジスタDに格納する。そして、ステップS302で、読出しデータのキーコードKCに対してシフトデータDを加算することにより音高変換し、元のルーチンに復帰する。

【0055】図9のベースパターンについての音高変換処理では、コードパターンのときと同様に、まず、ステップS41で、ベースパターンの読出しデータのキーコードKCをSRT(PTN)で音高シフトしてレジスタNTに格納し、ステップS42で、分類テーブルのデータATBL(STP(PTN), NT)をレジスタATRBに格納する。

【0056】次に、ステップS43で、ATRB=c(コード音でc1, c2, …c jの全て)のときATに“5”をセットし、ATRB=s(スケール音)のときATに“6”をセットし、ATRB=n(ノンスケール音)のときATに“7”をセットする。

【0057】そして、コードパターンのときと同様に、ステップS44で読出しデータのキーコードKCを指定和音の根音RTで音高シフトしてそのノートコードをレジスタNTに格納し、ステップS45で、ATに応じて選択したノート変換テーブルから、指定和音の種類TPと上記ノートコードNTに対応するシフトデータNTT(AT, TP, NT)を読み出し、これをレジスタDに格納する。そして、ステップS46で、読出しデータのキーコードKCに対してシフトデータDを加算することにより音高変換し、元のルーチンに復帰する。

【0058】なお、この実施例では、ソースパターンのキーコードKCがその属性に対応する指定和音の根音と種類で決められるスケールの属性になるようにシフトデータDを加算することによって、最終的なキーコードに対する根音シフトを必要としないので、伴奏パターンの音高がパターン全体的に上下することがなく、自然な流れになる。

【0059】以上のように、伴奏パターンの読出しデータのキーコードは、そのノートコードと伴奏パターン(ソースパターン)の和音の種類とに基づいてコード音、スケール音またはノンスケール音の何れかに分類され、さらに、この分類結果に応じたノート変換テーブルに基づいて音高変換される。なお、図5のノート変換テーブルの例からわかるように、コード音はコード音に変換され、スケール音はコード音またはスケール音に変換され、ノンスケール音はスケール音またはノンスケール

音に変換される。

【0060】このように、伴奏パターンのキーコードを、コード音、スケール音およびノンスケール音などのような和音の種類に応じたスケールに対する音名の属性に応じて分類し、それぞれの属性に応じたノート変換テーブルにより指定和音においても同じ属性になるように音高変換するようにしているので、異なる伴奏パターンを同じノート変換テーブルで変換しても、音楽的にそれぞれの伴奏パターンに適した音高に変換される。

【0061】このため、伴奏パターン(ソースパターン)毎にノート変換テーブルを作成する必要がないばかりか、ユーザが作成した伴奏パターンでも同じノート変換テーブルで音高変換を行うことができる。

【0062】また、属性がコード音であるキーコードに対して、ソースパターンの和音の根音と指定和音の根音とが所定範囲以上離れている場合に、同じコード音でも別の同類の属性に変換して最寄りの音程に変換するようにしているので、根音の差によるパターンのシフトが生じず、音域制限をしなくてもよい。このため、音域制限によるオクターブの折り返し等がなくなる。

【0063】

【発明の効果】以上説明したように本発明の自動伴奏装置によれば、和音に応じたスケールに対する音名の属性に基づいて、基伴奏パターンのキーコードを該基伴奏パターンの和音と入力される指定和音とに応じたキーコードに音高変換して自動伴奏を行い、この音高変換の際に、指定和音の根音と基伴奏パターンの和音の根音との差が予め定められた範囲以上のときは、基伴奏パターンのキーコードを、指定和音における属性が基伴奏パターンの和音における属性に近かつ異なるキーコードとなるように音高変換するようにしたので、基伴奏パターンのキーコードを基伴奏パターンの種類に係わらず音楽的に最適なものに変換でき、各伴奏パターンの種類に応じたソースパターン毎にノート変換テーブルを作成する必要がない。また、指定和音の根音と基伴奏パターンの根音とが大きく離れている場合でも、根音の差が変換後のキーコードの音域をシフトさせることが無くなり、音域制限を必要としない。

【図面の簡単な説明】

【図1】本発明実施例の自動伴奏装置を適用した電子楽器のブロック図である。

【図2】実施例における分類テーブルの一例を示す図である。

【図3】実施例における属性変換テーブルを示す図である。

【図4】実施例における属性変換の一例を示す図である。

【図5】実施例におけるノート変換テーブルの一例を示す図である。

【図6】実施例におけるメインルーチンのフローチャー

トである。

【図7】実施例における割込み処理のフローチャートである。

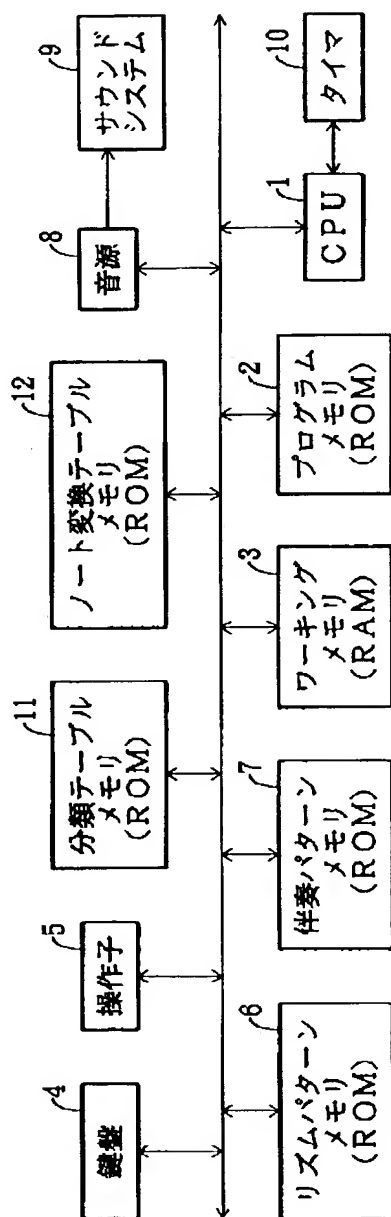
【図8】実施例におけるコードパターンについての音高変換処理のフローチャートである。

【図9】実施例におけるベースパターンについての音高変換処理のフローチャートである。

【符号の説明】

- | | |
|----|--------------|
| 1 | CPU |
| 2 | プログラムメモリ |
| 3 | ワーキングメモリ |
| 7 | 伴奏パターンメモリ |
| 11 | 分類テーブルメモリ |
| 12 | ノート変換テーブルメモリ |

【図1】



【図2】

ATBL (STP, NT) 分類テーブル

STP ↓		音名 (NT)											
		C	C#	D	D#	E	F	F#	G	G#	A	A#	B
和音種類	1	C#mJ	c4	n	s	n	c2	n	n	c5	n	s	n

	...	C#mJ7(9)	c4	n	c1	n	c2	n	n	c5	n	s	n

【図3】

(A) SFL (j, L)

差分 調性		L		
		-1~2	3~6, 6	5~2
J	1	1	3	2
	2	2	1	3
	3	3	2	1

(B) SFL (j, L)

差分 調性		L	
		-2~3	4~6, 6~3
J	4	4	5
	5	5	4

【図4】

(A)	差分	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C#mJ7(9)	0	c4	n	c1	n	c2	n	n	c5	n	s	n	c3
C#mJ7(9)	1	c3	c4	n	c1	n	c2	n	n	c5	n	s	n
D#mJ7(9)	2	n	c3	c4	n	c1	n	c2	n	n	c5	n	s
D#mJ7(9)	3	s	n	c3	c4	n	c1	n	c2	n	n	c5	n
E#mJ7(9)	4	n	s	n	c3	c4	n	c1	n	c2	n	n	c5
F#mJ7(9)	5	c5	n	s	n	c3	c4	n	c1	n	c2	n	n
F#mJ7(9)	±6	n	c5	n	s	n	c3	c4	n	c1	n	c2	n
G#mJ7(9)	-5	n	n	c5	n	s	n	c3	c4	n	c1	n	c2
G#mJ7(9)	-4	c2	n	n	c5	n	s	n	c3	c4	n	c1	n
A#mJ7(9)	-3	n	c2	n	n	c5	n	s	n	c3	c4	n	c1
A#mJ7(9)	-2	c1	n	c2	n	n	c5	n	s	n	c3	c4	n
B#mJ7(9)	-1	n	c1	n	c2	n	n	c5	n	s	n	c3	c4

(B)	差分	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C#mJ7(9)	0	c4	n	c1	n	c2	n	n	c5	n	s	n	c3
C#mJ7(9)	1	c3	c4	n	c1	n	c2	n	n	c5	n	s	n
D#mJ7(9)	2	n	c3	c4	n	c1	n	c2	n	n	c5	n	s
D#mJ7(9)	3	s	n	c3	c4	n	c1	n	c2	n	n	c5	n
E#mJ7(9)	4	n	s	n	c3	c4	n	c1	n	c2	n	n	c5
F#mJ7(9)	5	c4	n	s	n	c3	c4	n	c1	n	c2	n	n
F#mJ7(9)	±6	n	c4	n	s	n	c3	c4	n	c1	n	c2	n
G#mJ7(9)	-5	n	n	c4	n	s	n	c3	c4	n	c1	n	c2
G#mJ7(9)	-4	c3	n	n	c4	n	s	n	c3	c4	n	c1	n
A#mJ7(9)	-3	n	c3	n	n	c4	n	s	n	c3	c4	n	c1
A#mJ7(9)	-2	c2	n	c3	n	n	c4	n	s	n	c3	c4	n
B#mJ7(9)	-1	n	c1	n	c2	n	n	c5	n	s	n	c3	c4

【図5】

(A) NTT (3, TP, NT) ^{AT}コード音 (c4) 用 (コードバックシフト)

TP ↓		音名 (NT)												
		C	C#	D	D#	E	F	F#	G	G#	A	A#	B	
和音種類	1	Maj	0	-1	-2	-3	-4	7	6	5	4	3	2	1
											
											
	20	7-9	0	-1	-2	-3	-4	7	6	5	4	3	2	1

(B) NTT (5, TP, NT) コード音用 (ベース)

TP ↓			音名 (NT)											
			C	C#	D	D#	E	F	F#	G	G#	A	A#	B
和音種類	1	Maj	0	-1	2	1	0	-1	1	0	-1	-2	2	1
											
											
	20	7-9	0	1	0	-1	-2	2	1	0	-1	1	0	1

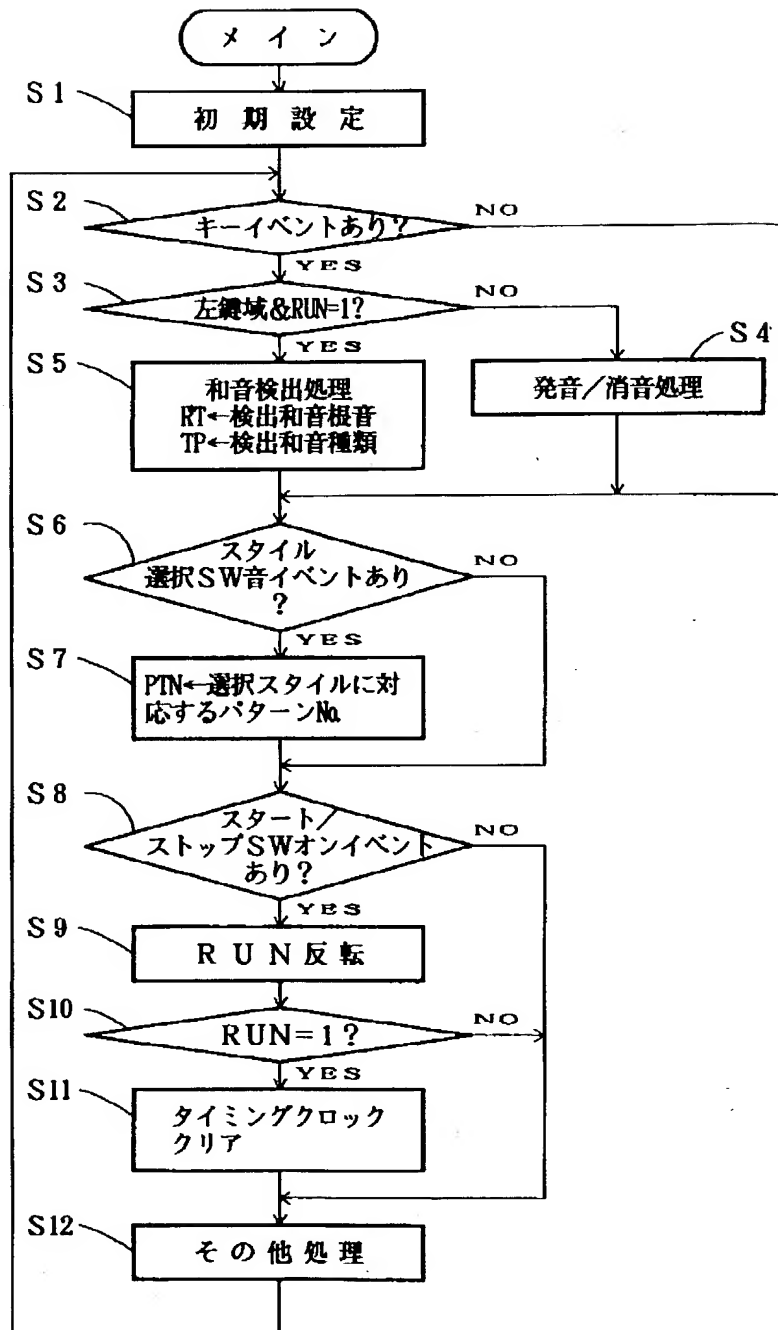
(C) NTT (8, TP, NT) スケール音 (s) 用

TP ↓			音名 (NT)											
			C	C#	D	D#	E	F	F#	G	G#	A	A#	B
和音種類	1	Maj	0	-1	0	-1	0	-1	1	0	-1	0	-1	1
	⋮	⋮	⋮											
	20	7-9	0	-1	0	-1	1	0	-1	0	0	-1	0	-1

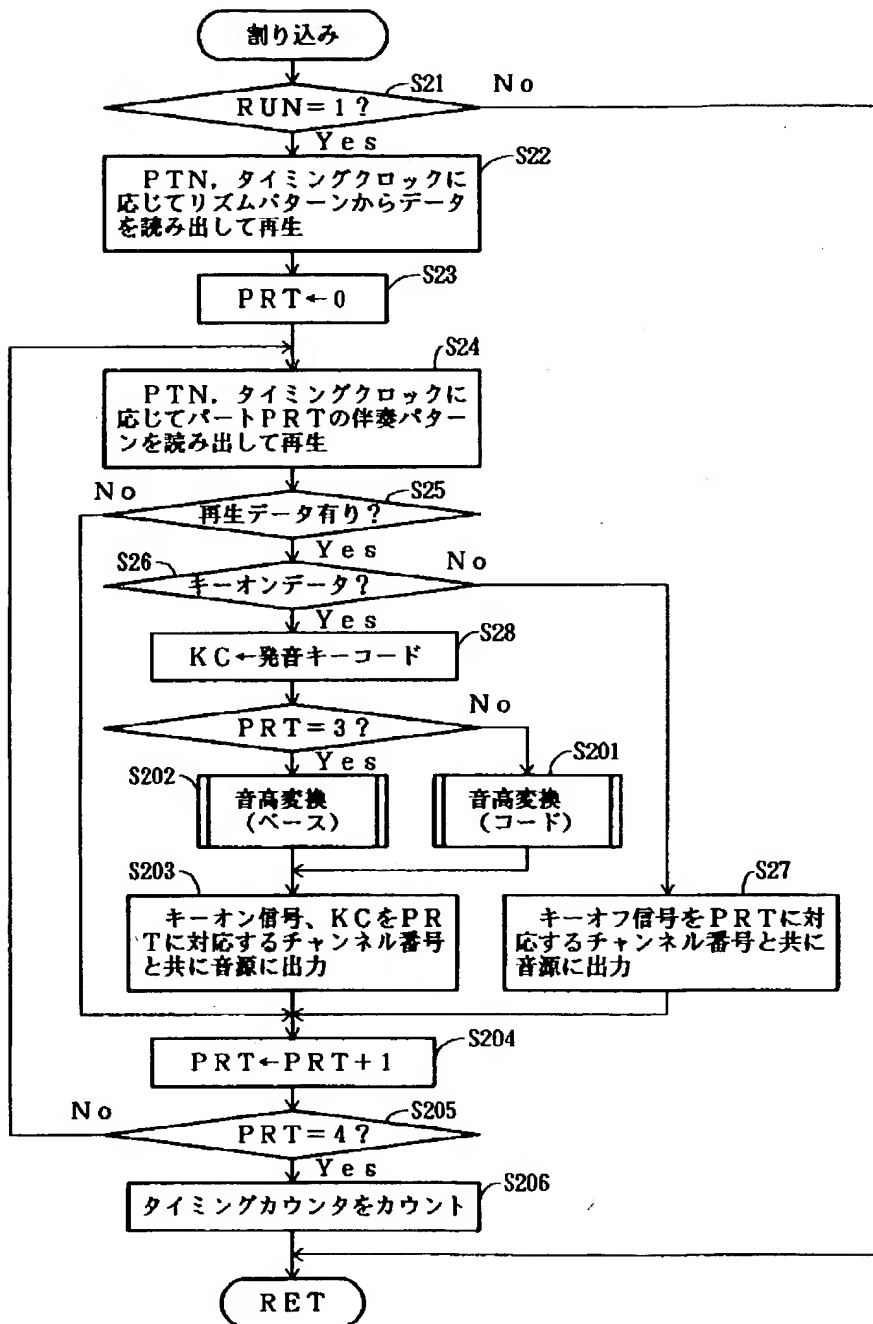
(D) NTT (7, TP, NT) ノンスケール音 (n) 用

TP ↓		音名 (NT)												
		C	C#	D	D#	E	F	F#	G	G#	A	A#	B	
和音種類	1	Maj	-1	0	0	0	-1	0	0	-1	0	0	0	0
	⋮	⋮					⋮							
	20	7-9	-1	0	-1	0	0	0	0	-1	0	0	-1	0

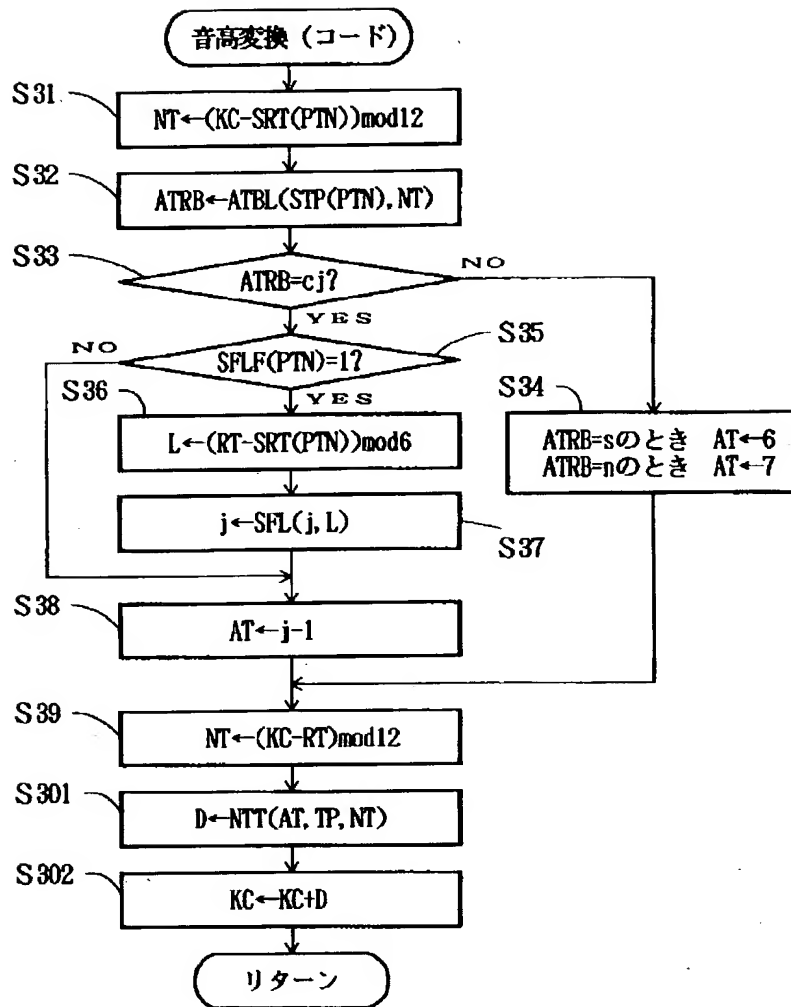
【図6】



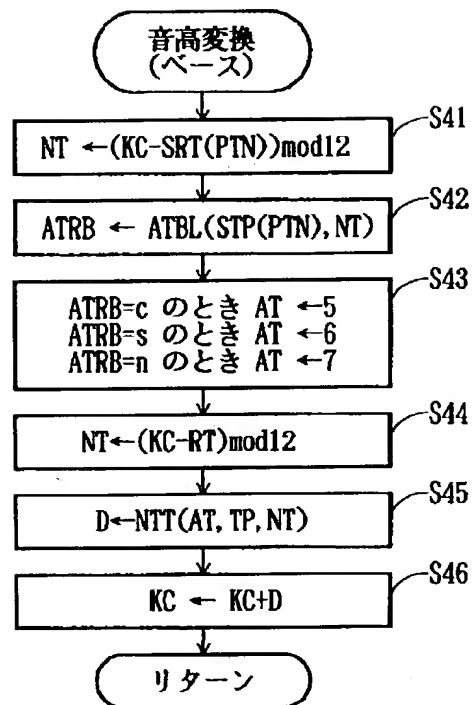
【図7】



【図8】



【図9】



PAT-NO: JP408185180A
DOCUMENT-IDENTIFIER: JP 08185180 A
TITLE: AUTOMATIC ACCOMPANIMENT DEVICE
PUBN-DATE: July 16, 1996

INVENTOR-INFORMATION:

NAME
KONDO, MASAO
NAKAZONO, HIROKI
ITO, SHINICHI

ASSIGNEE-INFORMATION:

NAME	COUNTRY
YAMAHA CORP	N/A

APPL-NO: JP06328824

APPL-DATE: December 28, 1994

INT-CL (IPC): G10H001/38, G10H001/00

ABSTRACT:

PURPOSE: To eliminate the labor for forming a note conversion table at every source pattern meeting the kinds of accompaniment patterns and to eliminate the limitation on compass with an automatic accompaniment device.

CONSTITUTION: Chord notes (c1 to c5), scale notes (s) and non-scale notes (n) are previously stored in a classification table as the attribute of the pitch names for the kinds of chords in accordance with the scales corresponding to the kinds of the chords. The note conversion table corresponding to the respective attributes is previously stored. The key chords in source patterns are classified by the attributes in accordance with their pitch names and the

kinds of the chords of the source patterns. The key chords are subjected to pitch conversion by the note conversion table corresponding to the classified attributes. The classified attributes are converted when the differences between the fundamental notes of the assigned chords and the fundamental notes of the chords of the source patterns are large. The pitch conversion is then so executed by the note conversion table corresponding to the converted attributes that the attributes in the assigned chords attain the different key chords approximate to the attributes in the chords of the fundamental accompaniment patterns.

COPYRIGHT: (C)1996,JPO